Simple Graphs: Connectedness, Trees

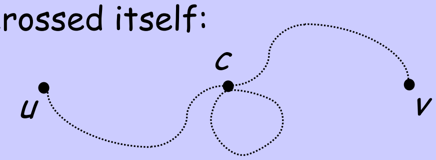
 March 7, 2007 lec 5W.1

Paths & Simple Paths

Lemma:

The *shortest* path between two vertices is simple!

Proof: Suppose path from u to v crossed itself:



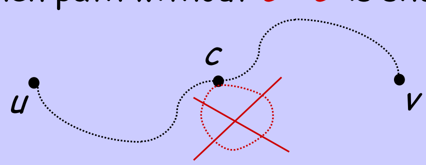
 March 7, 2007 lec 5W.2

Paths & Simple Paths

Lemma:

The *shortest* path between two vertices is simple!

Then path without $c \cdots c$ is shorter:



 March 7, 2007 lec 5W.3

Connected Graphs

A *connected* graph: there is a path between every two vertices.

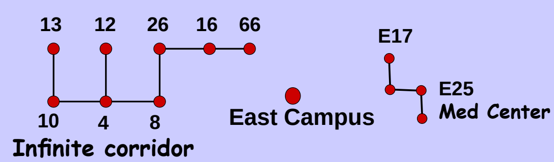
 March 7, 2007 lec 5W.4

Connected Components

Every graph consists of separate connected pieces (subgraphs) called *connected components*

 March 7, 2007 lec 5W.5

Connected Components



3 connected components

The more connected components, the more "broken up" the graph is.

 March 7, 2007 lec 5W.6

Connected Components

The *connected component* of vertex v:

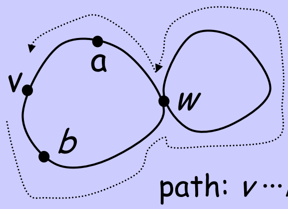
$\{w \mid v \text{ and } w \text{ are connected}\}$

Connected Components

So a graph is connected iff it has only 1 connected component

Cycles

A *cycle* is a path that begins and ends with same vertex

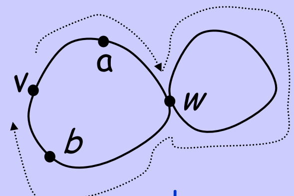


path: $v \cdots b \cdots w \cdots w \cdots a \cdots v$

also: $a \cdots v \cdots b \cdots w \cdots w \cdots a$

Cycles

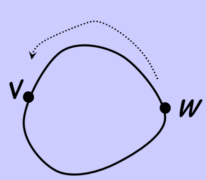
A *cycle* is a path that begins and ends with same vertex



also: $a \cdots w \cdots w \cdots b \cdots v \cdots a$

Simple Cycles

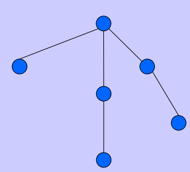
A simple *cycle* is a cycle that doesn't cross itself



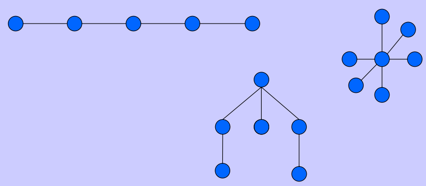
path: $v \cdots w \cdots v$ also: $w \cdots v \cdots w$

Trees

A *tree* is a connected graph with no cycles.



More Trees



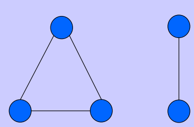
Other Tree Definitions

- A tree is a graph with a *unique* path between any 2 vertices.
- A tree is a connected graph with *n* vertices and *n* – 1 edges.
- A tree is an *edge-minimal* connected graph.

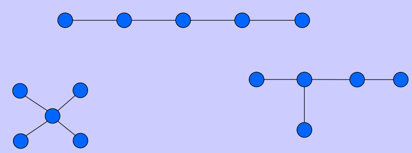
Be careful with these definitions

Is a tree simply a graph with *n* vertices and *n* – 1 edges?

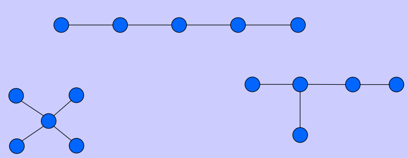
NO:



Some trees with five vertices



Some trees with five vertices

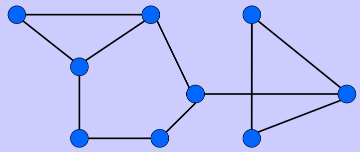


Exercise: Prove that all trees with five vertices are isomorphic to one of these three.

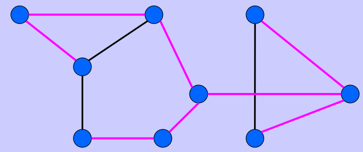
Spanning Trees

A *spanning tree*: a subgraph that is a tree on all the vertices.

Spanning Trees

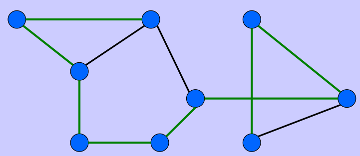


Spanning Trees



a spanning tree

Spanning Trees



another spanning tree
(can have many)

Spanning Trees

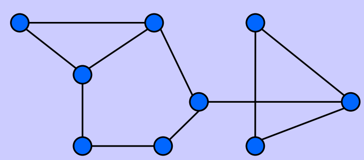
A *spanning tree*: a subgraph that is a tree on all the vertices.

Always exists: find *minimum edge-size*, connected subgraph on all the vertices.

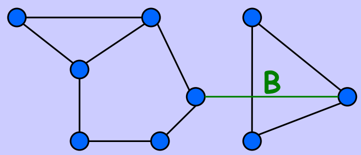
CONNECTEDNESS

An edge is a *cut edge* if removing it from the graph *disconnects* two vertices.

Cut Edges

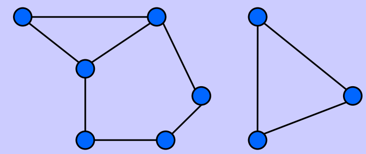


Cut Edges



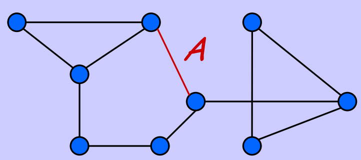
B is a cut edge

Cut Edges



deleting *B* gives two components

Cut Edges



A is *not* a cut edge

Cut Edges and Cycles

Lemma: An edge is a cut edge *iff* it is not traversed by a simple cycle.

Proof : problem set

Cut Edges

Fault-tolerant design:

- In a tree, every edge is a cut edge (bad)
- In a mesh, no edge is a cut edge (good; 2-connected)
- Tradeoff edges for failure tolerance

k-Connectedness

Def: k-connected iff need to delete *k* edges to disconnect.

k-Connectedness

Def: k-connected iff remains connected when any k-1 edges are deleted.

 March 7, 2007 lec 5W.32

k-Connectedness

Example:

K_n is (n-1)-connected

 March 7, 2007 lec 5W.33

Team Problems

Problems
1—3

 March 7, 2007 lec 5W.34